Archived:Basic PySymbian app - series3

Aquivado: Este artigo foi arquivado, pois o conteúdo não é mais considerado relevante para se criar soluções comerciais atuais. Se você achar que este artigo ainda é importante, inclua o template {{ForArchiveReview|*escreva a sua justificativa*}}.

All PySymbian articles have been archived. PySymbian is no longer maintained by Nokia and is not guaranteed to work on more recent Symbian devices. It is not possible to submit apps to Nokia Store.

Introduction

The first two articles of this series focused on application structure and a user friendly application. First Article explained how to design a application and the second one explained how to make a application which is user controlled.

In the previous article we all must have heard two terms very frequently they are

callback functions and

objects

This article will describe these terms. Callback functions and Objects are a very important terms when we want to develop applications in Python. Callback functions are not the same as normal functions and the difference will be emphasized in next paragraphs.

Callback Function

Callback function is more or less same as a normal function. Callback functions are defined similarly as a normal function. There is not a much of technical difference between a callback function and normal function but callback functions are rather used for some specific purpose. Generally a Callback function is called by a function to respond to a specific event for example in PySymbian when we choose a menu or decide to quit an application then a callback function is called. The code below explains the callback function:

```
import appuifw, e32
 appuifw.app.menu = [(u"menu1", callbck_1),
                     (u"menu2", callbck_2),
                     (u"menu3", callback_3),
                     (u"menu4", callback_4)]
```

Another key term of great importance is known as binding. Attaching a event with a function is called binding. Certain objects in PySymbian like Canvas have a bind() function so that a callback function is attached a to a specific event.

A callback function and a normal function differ only in one way when calling a callback function the parentheses are not used. Look at the code below:

```
def quit():
   print u"exit key pressed"

appuifw.app.exit_key_handler = quit # calling a callback function

quit() #calling a normal function
```

Objects

Object handling is very easy in python. Objects had the job of holding the variables and functions together that manipulate the objects. variables and functions are having many things in common so its not good to keep the variables and functions different. Objects are generally very much useful in large and complex applications where its good to divide the whole application into small, small parts. with the code below we can explain the objects concept in detail:

```
app_lock = e32.Ao_lock()

app_lock.wait()

app_lock.signal()
```

in the code above a object of module e32 i.e. **Ao_lock** is assigned to the variable **app_lock** and following to that the variable can be assigned to use the functions of the objects using the dot notation. As we are using the two functions wait and signal with the variable app_lock.

In simple language we can say that objects are modules inside modules.

Related Links

- Archived:Basic PySymbian application - series1
- Archived:Basic PySymbian app - series2

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.